



LAWRENCE  
LIVERMORE  
NATIONAL  
LABORATORY

# Improving Conservation for First-Order System Least Squares Finite-Element Methods

J. H. Adler, P. S. Vassilevski

September 6, 2012

Symposium in honor of Raytcho Lazarov's 40 years research  
in Computational Methods and Applied Mathematics  
Sozopol, Bulgaria  
June 7, 2013 through June 8, 2013

## **Disclaimer**

---

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

# IMPROVING CONSERVATION FOR FIRST-ORDER SYSTEM LEAST SQUARES FINITE-ELEMENT METHODS\*

J. H. ADLER<sup>†</sup> AND P. S. VASSILEVSKI<sup>‡</sup>

**Abstract.** The first-order system least-squares (FOSLS) finite element method for solving partial differential equations has many advantages, including the construction of symmetric positive definite algebraic linear systems that can be solved efficiently with multilevel iterative solvers. However, one drawback of the method is the potential lack of conservation of certain properties. One such property is conservation of mass. This paper describes a strategy for achieving mass conservation for a FOSLS system by changing the minimization process to that of a constrained minimization problem. In addition, an overlapping Schwarz process is used to make this strategy more robust and not add too much computational overhead to solving the system.

**Key words.** conservation, first-order system least squares, finite elements, domain decomposition

**AMS subject classifications.** 65F10, 65N20, 65N30

**1. Introduction.** The first-order system least squares (FOSLS) approach is a finite-element discretization which solves a system of linear partial differential equations (PDEs) by minimizing the  $L^2$  norm of the residual of the PDE [11, 12, 28, 29, 14]. Least-square finite-element methods, in general, have several nice properties and have been used on a wide variety of problems, e.g. [4, 5, 6, 8, 13, 27, 32]. One advantage is that they yield symmetric positive definite algebraic systems, which are amenable to multilevel techniques. This is true for any PDE system, including systems like Stokes where a mixed finite-element method would yield a saddle-point problem and an indefinite linear system [9]. Another advantage is that they yield sharp and reliable a posteriori estimates [3]. This is useful for implementing adaptive local refinement techniques, which allow the approximations to be resolved more accurately in regions of higher error [10, 17].

A disadvantage of the least-squares methods noted in the literature is a loss of conservation for certain properties in a given system. For instance, the Stokes' or Navier-Stokes' system contains an equation for the conservation of momentum and one for the conservation of mass [18, 19]. Since the least-square principle minimizes both equations equally, both quantities are only conserved up to the error tolerance given for the simulation. Attempts to improve the conservation of mass would result in a loss of accuracy in the conservation of momentum. Despite this, in several applications, conservation of a certain quantity is considered essential to capturing the true physics of the system. For instance, in electromagnetic problems, such as magnetohydrodynamics (the treatment of plasmas as charged fluids), loss of accuracy in the solenoidal constraint of the magnetic field,  $\nabla \cdot \mathbf{B} = 0$ , can lead to instabilities in the system [2, 7].

In this paper, we consider methods for improving the conservation of a divergence constraint, such as mass conservation, in a system, using the FOSLS finite-element

---

\*Submitted to "Springer Proceedings in Mathematics & Statistics", a volume dedicated to the 70th birthday of Raytcho Lazarov.

<sup>†</sup>Department of Mathematics, Tufts University, Medford, MA 02155 (james.adler@tufts.edu).

<sup>‡</sup>Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-560, Livermore, CA 94551, USA (panayot@llnl.gov). The work of the second author was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

method. There are many ways to improve the accuracy of mass conservation in such systems, including adaptive refinement to increase the spatial resolution of the discretization [4, 6], higher temporal accuracies or higher-order elements for time-dependent problems [30], using divergence-free finite-element spaces [5, 1, 15, 16], and reformulating the first-order system into a more conservative one [21]. In addition, an alternative approach called FOSLL\* [25, 26] has been developed, in which an adjoint system is considered, and the error is minimized in the  $L^2$  norm directly. This has been shown to improve the divergence-constraint conservation in incompressible fluid flow and electromagnetic problems. In this paper, we discuss an approach that simply corrects the solution approximated by the FOSLS discretization so that it conserves the given quantity. The goal is keep the discretization as is, preserving all of the special properties of the least-squares minimization, while still obtaining the appropriate conservation. As a result, the a posteriori error estimates and the simple finite-element spaces can still be used. More specifically, the aim of this paper is to show that it is possible to conserve a certain quantity in the least-squares finite element setting by using a local subdomain correction post-processing scheme at relatively little extra cost.

The paper is outlined as follows. In Section 2, we consider the FOSLS discretization applied to a Poisson problem and show how the scheme can result in a type of “mass loss”. Section 3 investigates a way of transforming the minimization principle into a constrained minimization problem and investigates what types of constraints are possible. Next, in Section 4, a local subdomain and coarse-grid correction solver is used to make the method more robust. This uses an overlapping Schwarz (Vanka-like) smoother with a coarse grid correction to solve the constrained problem [33, 34, 35]. Finally, concluding remarks and a discussion of future work is given in Section 5.

**2. First-Order System Least-Squares.** To illustrate the FOSLS finite-element method, consider a PDE system that is first put into a differential first-order system of equations, denoted by  $Lu = f$ . Here,  $L$  is a mapping from an appropriate Hilbert space,  $\mathcal{V}$ , to an  $L^2$  product space. In many contexts,  $\mathcal{V}$  is chosen to be an  $H^1$  product space with appropriate boundary conditions.

This minimization is written as

$$(2.1) \quad u_* = \arg \min_{u \in \mathcal{V}} G(u; f) := \arg \min_{u \in \mathcal{V}} \|Lu - f\|_0^2,$$

where  $u_*$  is the solution in an appropriate  $H^1$  space. The minimization results in the weak form of the problem:

Find  $u_* \in \mathcal{V}$  such that

$$(2.2) \quad \langle Lu_*, Lv \rangle = \langle f, Lv \rangle \quad \forall v \in \mathcal{V},$$

where  $\langle \cdot, \cdot \rangle$  is the usual  $L^2$  inner product on the product space,  $(L^2)^k$ , for  $k$  equations in the linear system. If the following properties of the bilinear form,  $\langle Lu, Lv \rangle$  are assumed,

$\exists$  constants,  $c_1$  and  $c_2$ , such that

$$(2.3) \quad \text{continuity} \quad \langle Lu, Lv \rangle \leq c_2 \|u\|_{\mathcal{V}} \|v\|_{\mathcal{V}} \quad \forall u, v \in \mathcal{V},$$

$$(2.4) \quad \text{coercivity} \quad \langle Lu, Lu \rangle \geq c_1 \|u\|_{\mathcal{V}}^2 \quad \forall u \in \mathcal{V},$$

then, by the Riesz Representation Theorem, this bilinear form is an inner product on  $\mathcal{V}$  [24]. In addition, these properties imply the existence of a unique solution,  $u_* \in \mathcal{V}$ , for the weak problem (2.2). Here,  $c_1$  and  $c_2$  depend only on the operator,  $L$ , and the domain of the problem. They are independent of  $u$  and  $v$ .

Next,  $u_*$  is approximated by restricting (2.1) to a finite-dimensional space,  $\mathcal{V}^h \subseteq \mathcal{V}$ , which leads to (2.2) restricted to  $\mathcal{V}^h$ . Since  $\mathcal{V}^h$  is a subspace of  $\mathcal{V}$ , the discrete problem is also well-posed. Choosing an appropriate basis,  $\mathcal{V}^h = \text{span}\{\Phi_j\}$ , and restricting (2.2) to this basis, yields an algebraic system of equations involving the matrix,  $A$ , with elements

$$(A)_{ij} = \langle L\Phi_j, L\Phi_i \rangle.$$

It has been shown that, in the context of a symmetric positive definite (SPD)  $H^1$ -equivalent bilinear form restricted to a finite element subspace, a multilevel technique exists that yields optimal convergence to the linear system [12].

**2.1. Sample Problem and Loss of Conservation.** To illustrate possible losses in conservation, consider the convection-diffusion equation for unknown  $p$  in two dimensions,

$$(2.5) \quad -\nabla \cdot D\nabla p + \mathbf{r} \cdot \nabla p + cp = f,$$

with  $D$  an SPD matrix that could depend on the domain and  $\mathbf{r}$  a vector and  $c$  a positive constant, respectively. In order to make the system first-order, a new variable,  $\mathbf{u} = D\nabla p$ , is introduced. The resulting FOSLS system becomes,

$$(2.6) \quad -\nabla \cdot \mathbf{u} + \mathbf{D}^{-1}\mathbf{r} \cdot \mathbf{u} + cp = f,$$

$$(2.7) \quad \nabla \times D^{-1}\mathbf{u} = 0,$$

$$(2.8) \quad D^{-1/2}\mathbf{u} - D^{1/2}\nabla p = 0.$$

Here, a scaling on  $D$  is performed to allow the resulting discrete system to be better conditioned and, thus, more amenable to multigrid methods. Also, the extra curl equation is introduced so that the weak system is continuous and coercive and, therefore,  $H^1$  equivalent [11, 12]. For simplicity, let  $D = 1$ ,  $\mathbf{r} = \mathbf{0}$ , and  $c = 0$ . Then, the following functional is minimized,

$$\mathcal{G} = \|\nabla \cdot \mathbf{u} + f\|_0^2 + \|\nabla \times \mathbf{u}\|_0^2 + \|\mathbf{u} - \nabla p\|_0^2.$$

The resulting discrete system is

$$A\mathcal{U} = b,$$

where  $\mathcal{U} = (\mathbf{u}, p)^T$ . When minimizing this functional, equal weight is given to each term in the system. Therefore, if better accuracy is needed on a certain term, such as the divergence constraint, accuracy is lost in the other portions. In many applications, however, exact conservation of certain terms is important for developing an accurate model of a physical system. For instance, one may want to conserve the “mass” of the system. This is defined as

$$(2.9) \quad \int_{\Omega} -\nabla \cdot \mathbf{u} \, d\Omega = \int_{\Omega} f \, d\Omega.$$

In other words, the amount of flow in or out of the system is equal to the flow contributed by the source (this has more physical meaning in a system like Stokes, where we assume  $\nabla \cdot \mathbf{u} = 0$  [18]). However, since the part of the functional concerned with this property is only minimized to a certain degree (i.e., truncation error of the scheme at best), this can not be satisfied exactly. Another issue concerns the fact that in many applications of the FOSLS finite-element method the same order of polynomials is chosen as the basis for every unknown in the discrete space. For instance, linear functions are chosen to approximate both  $\mathbf{u}$  and  $p$ . As a result, in trying to satisfy the term  $\mathbf{u} - \nabla p = 0$ , one is trying to match linears with the gradient of linears, or constants. This is not approximated very well and accuracy is lost. As a result the conservation property is also lost. Choosing higher-order elements does remedy this to some extent, especially in two dimensions. However, using higher-order elements increases the complexity of the discrete system and the grid-hierarchy in a multigrid scheme, making the systems harder to solve. In addition, the effect of higher-order elements is lessened when going to three dimensions [20, 22, 30].

To improve on this, here, the idea of adding the mass conservation as a constraint to the system is considered. Thus, instead of just minimizing the FOSLS functional, the functional is minimized subject to a constraint. This constraint enforces the desired mass conservation, while still allowing the FOSLS functional to be minimized as usual, thus retaining its nice properties. Next, several approaches for implementing this constraint are described.

**3. Constrained FOSLS.** To enforce the constraint mentioned above, a Lagrange multiplier,  $\lambda$ , is introduced and the FOSLS system is augmented as follows:

$$(3.1) \quad \begin{pmatrix} A & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} \mathcal{U} \\ \lambda \end{pmatrix} = \begin{pmatrix} b \\ g \end{pmatrix}.$$

Here,  $A$  and  $\mathcal{U}$  are as before for the FOSLS discretization,  $\lambda$  is the Lagrange multiplier, and  $C$  is a finite-element assembly of the constraint, in this example  $-\nabla \cdot \mathbf{u} = f$ . Two possible ways to construct  $C$  are considered. For the rest of the paper, we consider a triangulation of a mesh in two dimensions,  $\mathcal{T}_h$ , with grid spacing  $h$ . In addition, consider the polynomial spaces of order  $k$  defined on this triangulation as,  $\mathcal{P}_k$ . The following notation is used for matrices and spaces:

**DEFINITION 3.1.** Let  $\Phi_j \in [\mathcal{P}_{k_1}]^2$  be a vector. Let  $q_i \in \mathcal{P}_{k_2}$  and  $r_i \in \nabla \cdot [\mathcal{P}_{k_1}]^2$  be scalars. Then define the following matrices:

$$\tilde{B}_{ij} = \langle -\nabla \cdot \Phi_j, q_i \rangle,$$

$$\Lambda \Rightarrow \Lambda_{ij} = \langle -\nabla \cdot \Phi_j, -\nabla \cdot \Phi_i \rangle,$$

and right-hand sides:

$$\tilde{g} = \langle f, q_i \rangle, \quad g = \langle f, -\nabla \cdot \Phi_i \rangle.$$

**3.1. “Galerkin Constraint”.** Letting  $C = \tilde{B}$ , a standard Galerkin-type construction of the divergence constraint is obtained. It should be noted that the order of the polynomials for the constraints,  $k_2$ , can be different than the order for the FOSLS

unknowns,  $k_1$ , and, in fact, should be chosen to have less degrees of freedom so as not to over-constrain the system. The pairs chosen in this paper are quadratics-linears ( $\mathcal{P}_2 - \mathcal{P}_1$ ), quadratics-constants ( $\mathcal{P}_2 - \mathcal{P}_0$ ), and linears-constants ( $\mathcal{P}_1 - \mathcal{P}_0$ ). In this context,  $\mathcal{U} \in [\mathcal{P}_{k_1}]^3$  and  $\lambda \in \mathcal{P}_{k_2}$ . The resulting system is:

$$(3.2) \quad \begin{pmatrix} A & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} \mathcal{U} \\ \lambda \end{pmatrix} = \begin{pmatrix} b \\ \tilde{g} \end{pmatrix}.$$

**3.2. “Least-Squares Constraint”.** To keep faith with the FOSLS methodology, a constraint is proposed that is of the same form as that is used in the FOSLS discretization, namely, letting  $C = \Lambda$ . This allows the same finite-element spaces for the FOSLS unknowns to be used for the Lagrange multiplier. The system is then,

$$(3.3) \quad \begin{pmatrix} A & \Lambda \\ \Lambda & 0 \end{pmatrix} \begin{pmatrix} \mathcal{U} \\ \lambda \end{pmatrix} = \begin{pmatrix} b \\ g \end{pmatrix},$$

where  $\mathcal{U} \in [\mathcal{P}_{k_1}]^3$  and  $\lambda \in [\mathcal{P}_{k_1}]^2$ .

As is shown below, the system that needs to be solved in the least-squares constraint approach may not be well-conditioned. However, one can construct the constraint matrix  $C$  in such a way that it can be decomposed into a form which is much easier to solve. For instance, decompose  $\Lambda = B^T B$  (and, thus, the system is rewritten as

$$(3.4) \quad \begin{pmatrix} A & B^T B \\ B^T B & 0 \end{pmatrix} \begin{pmatrix} \mathcal{U} \\ \lambda \end{pmatrix} = \begin{pmatrix} b \\ g \end{pmatrix}.$$

However, the construction of  $B$  is not trivial in many cases (see, however, subsection 3.3.1) and it is easier to work with  $\tilde{B}$  instead. If the system in the “Galerkin” approach is taken and modified, the following is obtained:

$$(3.5) \quad \begin{pmatrix} A & \tilde{B}^T \tilde{B} \\ \tilde{B}^T \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} \tilde{\mathcal{U}} \\ \tilde{\lambda} \end{pmatrix} = \begin{pmatrix} b \\ \tilde{B}^T \tilde{g} \end{pmatrix}.$$

As it turns out, due to the following lemma, it is reasonable to solve system (3.5) instead of system (3.4).

**LEMMA 3.2.** *Consider systems (3.2) and (3.5). Let  $A$ ,  $\tilde{B}$ ,  $\mathcal{U}$ ,  $\lambda$ ,  $\tilde{\lambda}$ ,  $g$ , and  $\tilde{g}$  be all defined as above in Definition 3.1, then,*

$$\tilde{\lambda} = \tilde{B}\lambda \text{ and } \tilde{\mathcal{U}} = \mathcal{U}.$$

*Proof.* First combine the two systems,

$$(3.6) \quad A\mathcal{U} + \tilde{B}^T \lambda = f,$$

$$(3.7) \quad \tilde{B}\mathcal{U} = \tilde{g},$$

$$(3.8) \quad A\tilde{\mathcal{U}} + \tilde{B}^T \tilde{B}\tilde{\lambda} = f,$$

$$(3.9) \quad \tilde{B}^T \tilde{B}\tilde{\mathcal{U}} = \tilde{B}^T \tilde{g}.$$

Next, multiply equation (3.7) on the left by  $\tilde{B}^T$  and subtract the bottom two equations from the top two. Let  $e_{\mathcal{U}} = \mathcal{U} - \tilde{\mathcal{U}}$  and  $e_{\lambda} = \lambda - \tilde{B}\tilde{\lambda}$  to obtain,

$$\begin{aligned} Ae_{\mathcal{U}} + \tilde{B}^T e_{\lambda} &= 0, \\ \tilde{B}^T \tilde{B}e_{\mathcal{U}} &= 0. \end{aligned}$$

Since  $\tilde{B}^T$  is equivalent to a gradient operator, it can be shown that it is a one-to-one operator (since divergence and, thus,  $\tilde{B}$  is onto). Therefore,  $\tilde{B}e_\lambda = 0$  and the system becomes,

$$\begin{pmatrix} A & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} e_\mathcal{U} \\ e_\lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

which is the global ‘‘Galerkin’’ system, which is known to be invertible. As a result,  $e_\mathcal{U} = e_\lambda = 0$  and, more importantly,  $\mathcal{U} = \tilde{\mathcal{U}}$ , meaning solving either system results in the same solution.  $\square$

Therefore, (3.2) and (3.5) are both viable options for the constraint system. Next, each of these and some variations are tested to see which yield the best mass conservation with little extra computational work.

**3.3. Solvers.** To solve the constrained system, the conjugate-gradient (CG) method on the Schur complement is used [31]. Solving the system in this way yields the following set of equations:

$$\mathcal{U} = A^{-1}b - A^{-1}C^T\lambda,$$

$$CA^{-1}C^T\lambda = CA^{-1}b - g.$$

The second equation is solved for  $\lambda$  via CG and a backsolve is used to get the original  $\mathcal{U}$ . For the results presented here, a direct solver is used to compute  $A^{-1}$ , but in the future a multigrid solver, or whatever is used to solve the FOSLS system itself, will be substituted instead.

For the first-approach (3.2) and second (3.3) the system is solved exactly as described above. In the second approach,  $\Lambda = B^TB$ , where  $B$  represents the construction of  $\langle -\nabla \cdot \Phi_j, r_i \rangle$ , but where  $r = \nabla \cdot \Phi$  is in the divergence of the space used for  $A$ , i.e  $\nabla \cdot \mathcal{P}_{k_1}$  as opposed to the full  $\mathcal{P}_{k_2}$ . As a result, the Schur complement equation becomes

$$(3.10) \quad B^TBA^{-1}B^TB\lambda = B^TBA^{-1}b - g.$$

This is badly conditioned as the system  $B^TB$  is equivalent to a  $-\nabla\nabla\cdot$  equation. However, to remedy this, the equation is multiplied on the left by  $BA^{-1}$ , resulting in

$$(BA^{-1}B^T)(BA^{-1}B^T)B\lambda = (BA^{-1}B^T)BA^{-1}b - (BA^{-1})g.$$

Notice that  $BB^T$  is equivalent to a  $-\nabla \cdot \nabla$  or Laplace system and thus,  $BA^{-1}B^T$  is well-conditioned. In addition, one only needs to solve for  $B\lambda$ . This system simplifies further by eliminating one of the  $BA^{-1}B^T$  blocks to obtain,

$$(3.11) \quad (BA^{-1}B^T)B\lambda = BA^{-1}b - (BA^{-1}B^T)^{-1}BA^{-1}g.$$

However, in (3.11), two solves of  $BA^{-1}B^T$  are required, increasing the number of iterations required to solve the system.

In addition, a problem with this approach is the construction of  $B$ . A simpler way is to construct  $\tilde{B}$  and use this instead to get system (3.5). This results in

$$(3.12) \quad \tilde{B}^T\tilde{B}A^{-1}\tilde{B}^T\tilde{B}\lambda = \tilde{B}^T\tilde{B}A^{-1}b - \tilde{B}^T\tilde{g}.$$



Multiplying on the left by  $\tilde{B}A^{-1}$  yields

$$(3.13) \quad \begin{aligned} (\tilde{B}A^{-1}\tilde{B}^T)(\tilde{B}A^{-1}\tilde{B}^T)\tilde{B}\lambda &= (\tilde{B}A^{-1}\tilde{B}^T)\tilde{B}A^{-1}b - (\tilde{B}A^{-1}\tilde{B}^T)\tilde{g} \\ (\tilde{B}A^{-1}\tilde{B}^T)\tilde{B}\lambda &= \tilde{B}A^{-1}b - \tilde{g}. \end{aligned}$$

This, however, is the same system obtained from (3.2) and, as shown in Lemma 3.2, results in the same solution for  $\mathcal{U}$ .

**3.3.1. Construction of  $B$ .** Despite being able to use the simpler construction,  $\tilde{B}$ , it is possible to construct  $B$  for the type of constraint considered here,  $\nabla \cdot \mathbf{u} = f$ . In fact, the matrix  $B$  is constructed locally using the simpler construction of  $\tilde{B}$ . Consider an element (triangle)  $T$  and let  $[\mathcal{P}_{k_1}]^2(T)$  be the vector polynomials of degree  $k_1$ . Next, consider the “Least-Squares” constraint, where the space of Lagrange multipliers,  $\lambda$ , is  $\nabla \cdot [\mathcal{P}_{k_1}]^2(T)$ , which is a subspace of  $[\mathcal{P}_{k_1-1}](T)$ . Let  $\{\varphi_s\}_{s=1}^l$  be the basis (restricted to  $T$ ) of  $[\mathcal{P}_{k_1-1}](T)$ . For  $k_1 = 2$ ,  $l = 3$  (since  $[\mathcal{P}_{k_1-1}](T) = [\mathcal{P}_1](T)$  – the space of linears). Also, let  $\{\Phi_i\}_{i=1}^n$  be the basis of  $[\mathcal{P}_{k_1}]^2(T)$ . Since  $\nabla \cdot \Phi_i \in \nabla \cdot [\mathcal{P}_{k_1}]^2(T) \subset [\mathcal{P}_{k_1-1}](T)$ ,

$$(3.14) \quad \nabla \cdot \Phi_i = \sum_{s=1}^l c_{i,s} \varphi_s = [\varphi_1, \dots, \varphi_l] \mathbf{c}_i,$$

for some coefficients  $\mathbf{c}_i = (c_{i,s}) \in \mathbb{R}^l$ . Therefore,

$$\left( \tilde{B}_T \right)_{i,s} = (\nabla \cdot \Phi_i, \varphi_s) = [(\varphi_s, \varphi_1), \dots, (\varphi_s, \varphi_l)] \mathbf{c}_i = \mathbf{e}_s^T M \mathbf{c}_i.$$

Here,  $\mathbf{e}_s \in \mathbb{R}^l$  is the  $s$ th unit coordinate vector and  $M = M_T$  is the element mass matrix coming from the space  $[\mathcal{P}_{k_1-1}](T)$ . In conclusion, the element matrix  $\tilde{B} = \tilde{B}_T = ((\nabla \cdot \Phi_i, \varphi_s))_{1 \leq i \leq n, 1 \leq s \leq l}$  admits the following form

$$\tilde{B}_T = M_T [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n].$$

For the entries  $(\nabla \cdot \Phi_i, \nabla \cdot \Phi_j) = (B_T^T B_T)_{ij}$ , using the representation (3.14) yields

$$(B_T^T B_T)_{ij} = (\nabla \cdot \Phi_i, \nabla \cdot \Phi_j) = \mathbf{c}_j^T ((\varphi_r, \varphi_s))_{r,s=1}^l \mathbf{c}_i = \mathbf{c}_j^T M_T \mathbf{c}_i = \left( \tilde{B}_T^T M_T^{-1} \tilde{B}_T \right)_{ij}.$$

Therefore,

$$B_T = M_T^{-\frac{1}{2}} \tilde{B}_T.$$

Thus,  $B$  is constructed relatively easily. Namely, over each element the local matrix,  $\tilde{B}_T$ , is built, which is the Galerkin finite-element construction of the divergence operator using  $P_{k_1} - P_{k_1-1}$  elements. Then,  $B_T = M_T^{-1/2} \tilde{B}_T$ , where  $M_T^{-1/2}$  is the mass matrix associated with the given element and  $P_{k_1-1}$ .

**3.4. Numerical Results.** In the following numerical tests, four approaches are considered:

- Method 1: Solve the “Galerkin” constraint system, (3.2), resulting in (3.13). Note, that this is the same as solving System (3.5) and simplifying the Schur complement system.
- Method 2: Solve the “Least-Squares” constraint system, (3.3), resulting in (3.10).

- Method 3: Solve the “Least-Squares” constraint system using the simpler construction, (3.5), resulting in (3.12).
- Method 4: Solve the “Least-Squares” constraint system, (3.4) with the simplified Schur complement system, (3.11).

Again,  $D = 1$ ,  $\mathbf{r} = \mathbf{0}$ , and  $c = 0$ . The right-hand side is chosen as  $f = 2\pi^2 \sin(\pi x) \sin(\pi y)$  so that the true solution is  $p = \sin(\pi x) \sin(\pi y)$ . The problem is solved on a unit square with homogeneous Dirichlet boundary conditions for  $p$ . The system is solved using the four approaches described above for a combination of the finite-element spaces,  $\mathcal{P}_2$ ,  $\mathcal{P}_1$ , and  $\mathcal{P}_0$ . The  $L^2$  norms of the errors of the solutions,  $p$  and  $\mathbf{u} = \nabla p$ , are shown in the following tables. Here,  $u_{err} = \|\mathbf{u} - \mathbf{u}^*\|_0 / \|\mathbf{u}^*\|_0$  and  $p_{err} = \|p - p^*\|_0 / \|p^*\|_0$  for the constrained system, where  $\mathbf{u}^*$  and  $p^*$  are the true solutions. The FOSLS functional,  $\mathcal{F} = \|\mathcal{L}\mathbf{u} - f\|_0$ , is given for both the unconstrained system,  $\mathcal{F}$ , and the constrained system,  $\mathcal{F}_c$ . In addition, the mass conservation (or mass loss) is shown as

$$m_L = \left| \int_{\Omega} -\nabla \cdot \mathbf{u} - f \, d\Omega \right|,$$

for the unconstrained FOSLS system as well as with the constraint,  $m_L^c$ . Finally, the number of iterations, it, needed in the CG algorithm to reduce the algebraic residual by  $10^{-8}$  are shown.

$k_1$	$k_2$	h	$m_L$	$m_L^c$	$\mathcal{F}$	$\mathcal{F}_c$	$u_{err}$	$p_{err}$	it
1	0	1/16	2.8e-2	4.9e-11	0.90	1.12	0.181	0.015	113
1	0	1/32	1.0e-2	2.9e-11	0.48	0.86	0.181	0.004	232
2	0	1/16	3.7e-5	5.7e-14	3.7e-2	3.7e-2	1.16e-3	1.40e-4	4
2	0	1/32	2.4e-6	3.0e-14	9.5e-3	9.5e-3	1.82e-4	1.73e-5	2
2	1	1/16	3.7e-5	1.9e-13	3.7e-2	3.7e-2	1.12e-3	1.38e-4	13
2	1	1/32	2.4e-6	3.4e-14	9.5e-3	9.5e-3	1.81e-4	1.72e-5	7

Table 3.1: (Method 1) Solve  $\tilde{B}A^{-1}\tilde{B}^T\lambda = \tilde{B}A^{-1}b - \tilde{g}$ . This approach is equivalent to using the “Galerkin” approach, (3.2), and the “Least-Squares” approach plus simplification of the Schur complement system on  $\tilde{B}$ , (3.5).

$k_1$	$k_2$	h	$m_L$	$m_L^c$	$\mathcal{F}$	$\mathcal{F}_c$	$u_{err}$	$p_{err}$	it
1	1	1/16	2.8e-2	5.1e-12	0.90	1.12	0.181	0.015	1,730
1	1	1/32	1.0e-2	2.1e-13	0.48	0.86	0.181	0.004	20,375
2	2	1/16	3.7e-5	1.5e-11	3.7e-2	9.8e-2	0.012	1.38e-4	1,100
2	2	1/32	2.4e-6	1.7e-12	9.5e-3	4.8e-2	4.94e-3	1.72e-5	4,319

Table 3.2: (Method 2) Solve  $\Lambda A^{-1}\Lambda\lambda = \Lambda A^{-1}b - g$ . This approach is equivalent to using the “Least-Squares” approach, but without splitting the constraint matrix and solving the full Schur complement system, (3.3).

**3.5. Discussion.** A couple of things to note are the fact that the first test yields some of the most optimal results. Method 2 attempts to solve the ill-conditioned  $\nabla\nabla\cdot$

$k_1$	$k_2$	h	$m_L$	$m_L^c$	$\mathcal{F}$	$\mathcal{F}_c$	$u_{err}$	$p_{err}$	it
1	1	1/16	2.8e-2	2.8e-12	0.90	1.12	0.181	0.015	1,600
1	1	1/32	1.0e-2	2.9e-11	0.48	0.86	0.181	0.004	15,268
2	1	1/16	3.7e-5	9.3e-13	3.7e-2	3.7e-2	1.16e-3	1.40e-4	15
2	1	1/32	2.4e-6	7.5e-14	9.5e-3	9.5e-3	1.82e-4	1.73e-5	4
2	2	1/16	3.7e-5	5.9e-14	3.7e-2	3.7e-2	1.15e-3	1.38e-4	12
2	2	1/32	2.4e-6	7.3e-14	9.5e-3	9.5e-3	1.81e-4	1.72e-5	6

Table 3.3: (Method 3) Solve  $\tilde{B}^T \tilde{B} A^{-1} \tilde{B}^T \tilde{B} \lambda = \tilde{B}^T \tilde{B} A^{-1} b - \tilde{B}^T \tilde{g}$ . This approach is equivalent to using the “Least-Squares” approach with the simpler construction of the constraint, but without splitting the constraint matrix and solving the full Schur complement system, (3.5).

$k_1$	$k_2$	h	$m_L$	$m_L^c$	$\mathcal{F}$	$\mathcal{F}_c$	$u_{err}$	$p_{err}$	it
1	1	1/16	2.8e-2	1.3e-10	0.90	1.12	0.181	0.015	84+134
1	1	1/32	1.0e-2	7.8e-10	0.48	0.86	0.181	0.004	146+307
2	2	1/16	3.7e-5	5.1e-10	3.7e-2	9.8e-2	0.012	1.38e-4	72+101
2	2	1/32	2.4e-6	1.7e-9	9.5e-3	4.8e-2	7.73e-3	1.72e-5	124+198

Table 3.4: (Method 4) Solve  $\Lambda A^{-1} \Lambda \lambda = \Lambda A^{-1} b - b$ . This approach is equivalent to using the “Least-Squares” approach, and using the simplification of the full Schur complement system using  $B$ , (3.11). Note, that since two solves of  $BA^{-1}B^T$  are required, the iterations for both solves are displayed in the last column of the table.

like system and, as is shown, requires too many iterations to be used reliably. Methods 3 and 4 improve on this, however, as they require extra solves in the solution process they require more work than in the first case.

In addition, only when a stable pair of elements with the constraint is used (i.e.  $\mathcal{P}_2 - \mathcal{P}_0$  or  $\mathcal{P}_2 - \mathcal{P}_1$ ) are the optimal results obtained. This results from the fact that only for the stable combinations is there enough room to minimize the FOSLS functional. All cases yield improved conservation as this is enforced directly. However, for the unstable pairings as the constraint is enforced, only a few possible solutions are allowed and, as a result, when the FOSLS functional is minimized, there is no longer enough room to minimize certain terms in the functional any more (such as  $\mathbf{u} - \nabla p = 0$ ). Thus, the best  $\mathbf{u}$  is not found. The solution has better mass conservation, but the approximation is not necessarily capable of minimizing the FOSLS functional. This can be seen by looking at the reduction in the error of  $\mathbf{u}$ . In all cases, the solution,  $p$ , is approximated well and the error is reduced with  $h$  as expected. However, for the unstable pairs, the gradient,  $\mathbf{u}$ , is not approximated well. Thus, the functional is no longer estimating the  $H^1$  error accurately and the a posteriori error estimator is lost. Therefore, the conclusion is that the constraint always needs to be chosen from a space which gives a stable finite-element pair with whatever unknowns from the FOSLS system that you wish to conserve. This requires considering an inf-sup condition for the FOSLS unknown and Lagrange multiplier pairs, but in many applications these pairs of spaces are well-known.

Alternatively, we may use for the constraints, test functions from a coarse subspace of a space that generally may not provide a stable fine-grid pair. For instance,

if the constraint matrix,  $\tilde{B}$ , is constructed using the “Galerkin-like” approach using the same polynomial space as the FOSLS system, the finite-element pairs are not stable. However, if this operator is restricted to a coarser space,  $H$ , and the Lagrange multiplier,  $\lambda_H$ , is chosen in that coarser space, stability is regained (assuming the coarse space is “coarse enough”). In the following results, this is tested using linears and quadratics. An interpolation operator is constructed via standard finite-element interpolation,  $Q_H$ , which takes DOF from a grid of size  $H$  and interpolates it to the fine-grid,  $h$ . Thus, the constrained system becomes,

$$(3.15) \quad \begin{pmatrix} A & \tilde{B}^T Q_H \\ Q_H^T \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} \mathcal{U} \\ \lambda_H \end{pmatrix} = \begin{pmatrix} b \\ Q_H^T \tilde{g} \end{pmatrix}.$$

$k_1$	$k_2$	h	H	$m_L$	$m_L^c$	$\mathcal{F}$	$\mathcal{F}_c$	$u_{err}$	$p_{err}$	it
1	1	1/8	1/4	5.3e-2	1.9e-12	1.65	1.70	0.222	0.056	21
1	1	1/16	1/8	2.8e-2	9.0e-12	0.90	0.91	0.132	0.013	27
1	1	1/16	1/4	2.8e-2	3.0e-11	0.90	0.90	0.134	0.015	17
1	1	1/32	1/16	1.0e-2	6.6e-13	0.48	0.48	0.053	0.003	25
1	1	1/32	1/8	1.0e-2	7.1e-12	0.48	0.48	0.053	0.004	17
1	1	1/32	1/4	1.0e-2	4.0e-12	0.48	0.48	0.053	0.004	13
2	2	1/8	1/4	5.5e-4	1.5e-11	0.14	0.15	0.008	0.001	31
2	2	1/16	1/8	3.7e-5	7.0e-13	3.7e-2	3.9e-2	1.10e-3	1.88e-4	28
2	2	1/16	1/4	3.7e-5	8.6e-13	3.7e-2	3.8e-2	1.11e-3	1.39e-4	22
2	2	1/32	1/16	2.4e-6	4.6e-13	9.5e-3	9.9e-3	1.79e-4	3.69e-5	22
2	2	1/32	1/8	2.4e-6	7.5e-14	9.5e-3	9.6e-3	1.79e-4	2.32e-5	17
2	2	1/32	1/4	2.4e-6	9.4e-14	9.5e-3	9.5e-3	1.80e-4	1.83e-5	16

Table 3.5: (Alternative Approach) Solve (3.15), where  $Q_H^T \tilde{B}$  is the “Galerkin” constraint on a coarser mesh.

As is seen in Table 3.5, using  $\mathcal{P}_1 - \mathcal{P}_1$  and  $\mathcal{P}_2 - \mathcal{P}_2$  pairs yields conservation and still allows the FOSLS functional to be minimized as expected. Thus, the solution,  $p$ , and it’s gradient,  $\mathbf{u}$ , are approximated well with only a handful of extra iterations needed.

#### 4. Locally Constrained FOSLS Correction.

**4.1. Overlapping Schwarz corrections.** Now, that it has been shown that augmenting the FOSLS system with a constraint gives better mass conservation with only a few extra iterations, a more robust local way of solving the problem is described here. An overlapping Schwarz process, as described in [35] (Section 9.5), is considered to break the constrained problem into smaller local problems. First consider that the FOSLS discrete system has been solved. In other words, no constraints are yet imposed. Then, the following post-processing step is performed. Let  $\{\Omega_i\}$  be an overlapping partition of  $\Omega$  into mesh subdomains (i.e., each  $\Omega_i$  is a union of fine-grid elements). Then correct the current solution  $\mathcal{U}$  with

$$\mathcal{U}_i \in V_h^0(\Omega_i) = \{ \mathbf{v} \in V_h : \text{supp}(\mathbf{v}_i) \subset \overline{\Omega_i} \},$$

by solving the locally constrained minimization problem for  $\mathcal{U}_i \in V_h^0(\Omega_i)$  and  $\lambda_i \in \mathcal{R}_i = \nabla \cdot V_h^0(\Omega_i)$  posed in  $\Omega_i$ :

$$\begin{aligned} a(\mathcal{U} + \mathcal{U}_i, \mathbf{v}_i) + (\lambda_i, \nabla \cdot \mathbf{v}_i) &= (F, \mathbf{v}_i), \text{ for all } \mathbf{v}_i \in V_h^0(\Omega_i), \\ (\nabla \cdot (\mathcal{U} + \mathcal{U}_i), \varphi) &= (f, \varphi) \text{ for } \varphi \in \mathcal{R}_i. \end{aligned}$$

Here, for the local space  $\mathcal{R}_i \equiv \nabla \cdot V_h^0(\Omega_i)$ , the local systems can be constructed as in Section 3.3.1. Likewise, a computational basis, based on QR or SVD, can be obtained as well. This is feasible if the domains  $\Omega_i$  are relatively small. Next, set  $\mathcal{U} := \mathcal{U} + \mathcal{U}_i$  and move onto the next subdomain  $\Omega_{i+1}$ .

After several loops over the Schwarz subdomains, a global coarse-space correction is performed. For this, a coarse space,  $\mathcal{R}_H \subset \nabla \cdot V_h$ , is needed with an explicit locally supported basis such that the pair  $(V_h, \mathcal{R}_H)$  is L.B.B.-stable. Alternatively, based on a coarse space,  $V_H \subset V_h$ , and coarser subdomains,  $\{\Omega_i^H\}$  (i.e., union of coarse elements in  $\mathcal{T}_H$ ), for the current approximation  $\mathcal{U} \in V_h$ , local coarse-space corrections,  $\mathcal{U}_i^H \in V_H^0(\Omega_i^H) = \{\mathbf{v}_H \in V_H : \text{supp}(\mathbf{v}_H) \subset \bar{\Omega}_i^H\}$ , are obtained by solving the local saddle-point problems for  $\mathcal{U}_i^H \in V_H^0(\Omega_i^H)$  and  $\lambda_i^H \in \mathcal{R}_i^H = \nabla \cdot V_H^0(\Omega_i^H)$  posed in  $\Omega_i^H$ :

$$\begin{aligned} a(\mathcal{U} + \mathcal{U}_i^H, \mathbf{v}_i^H) + (\lambda_i^H, \nabla \cdot \mathbf{v}_i^H) &= (F, \mathbf{v}_i^H), \text{ for all } \mathbf{v}_i^H \in V_H^0(\Omega_i^H), \\ (\nabla \cdot (\mathcal{U} + \mathcal{U}_i^H), \varphi) &= (f, \varphi) \text{ for } \varphi \in \mathcal{R}_i^H. \end{aligned}$$

Here, the coarse spaces can be constructed in a variational way by using standard interpolation and restriction operators for polynomial finite-element spaces. Finally, let  $\mathcal{U} := \mathcal{U} + \mathcal{U}_i^H$  and move onto the next coarse subdomain  $\Omega_{i+1}^H$ . The process can be applied recursively in a  $V$ -cycle iteration exploiting the above constrained overlapping Schwarz (Vanka-like) smoothing corrections [34]. For this paper, however, we consider only a two-level method with one global coarse space.

**4.2. Numerical Results.** To test the scheme described above in section 4.1, the ‘‘Galerkin’’-like constrained system, (3.2), is considered on subdomains and a coarse-grid. This system gave the most optimal results (fewer iterations and better mass conservation) and, therefore, appears to be the natural choice for performing the subdomain corrections. As described above, the standard FOSLS system is solved yielding,  $\mathcal{U}_0$ . Next, the finite-element triangulation of  $\Omega$  is divided into overlapping subdomains,  $\mathcal{T}_i$  of  $\Omega_i$ . The restriction of the FOSLS system,  $A$ , and the constraint equation,  $\tilde{B}$ , is formed by a simple projection onto the subdomains giving,  $A_i = P_i^T A P_i$  and  $B_i = Q_i^T \tilde{B} P_i$ . Here,  $P_i$  and  $Q_i$  are the natural injection operators of DOFs on  $\mathcal{T}_i$  to the original mesh,  $\mathcal{T}$ , for elements of  $\mathcal{P}_{k_1}$  and  $\mathcal{P}_{k_2}$ , respectively. Then, on each subdomain the Schur complement system of the error equations is solved as described above in section 4.1. Once all corrections on subdomains are updated, the system is projected onto a coarse grid,  $\mathcal{T}_H$ , where an update is again solved for. We use the standard finite-element interpolation operators to move between a coarse grid of size  $H$  to a fine grid of size  $h$ . We define these as  $P_H$  for  $\mathcal{P}_{k_1}$  and  $Q_H$  for  $\mathcal{P}_{k_2}$ . Note, that  $P_H$  is a block matrix of interpolation operators for each unknown in the FOSLS system. The transposes are used as restriction operators from fine grid to coarse grid. The algorithm is described below, letting  $M_s$  be the maximum number of subdomain smoothing steps and  $N_{sd}$  being the number of overlapping subdomains:

---



---

```

Solve FOSLS System:  $A\mathcal{U}_0 = b$ .
Compute Residuals:  $r_A = b - A\mathcal{U}_0$  and  $r_B = \tilde{g} - \tilde{B}\mathcal{U}_0$ .
Set  $\mathcal{U} = \mathcal{U}_0$  and  $\lambda = 0$ .
Perform Subdomain Smoothing Steps:
for  $s = 1$  to  $M_s$  do
    for  $i = 1$  to  $N_{sd}$  do
        Restrict Matrices and Residuals to Subdomains.
        Solve:  $\begin{pmatrix} A_i & B_i^T \\ B_i & 0 \end{pmatrix} \begin{pmatrix} \mathcal{U}_i \\ \lambda_i \end{pmatrix} = \begin{pmatrix} P_i^T r_A \\ Q_i^T r_B \end{pmatrix}$ .
        Update:  $\mathcal{U} = \mathcal{U} + P_i \mathcal{U}_i$  and  $\lambda = \lambda + Q_i \lambda_i$ .
        Recompute Residuals:  $r_A = b - A\mathcal{U} - \tilde{B}^T \lambda$  and  $r_B = \tilde{g} - \tilde{B}\mathcal{U}_0$ .
    end
end
Perform Coarse-Grid Correction:
Solve:  $\begin{pmatrix} P_H^T A P_H & P_H^T \tilde{B}^T Q_H \\ Q_H^T \tilde{B} P_H & 0 \end{pmatrix} \begin{pmatrix} \mathcal{U}_H \\ \lambda_H \end{pmatrix} = \begin{pmatrix} P_H^T r_A \\ Q_H^T r_B \end{pmatrix}$ .
Update:  $\mathcal{U} = \mathcal{U} + P_H \mathcal{U}_H$ .

```

---

The results for  $\mathcal{P}_2 - \mathcal{P}_0$  and  $\mathcal{P}_1 - \mathcal{P}_0$  pairs of elements for the FOSLS solution and the constraint variable are given in Table 4.1 using various grid spacings. The first set of results is given for the original FOSLS system with no constraint correction. The FOSLS functional is reduced by  $h^{k_1}$  as expected and it gives a good approximation of the reduction in error for both  $\mathbf{u}$  and  $p$ . However, the mass loss is rather large. Using quadratics improves the results but not exactly. The remaining blocks of data give the results using various numbers of smoothing steps and with or without coarse-grid corrections. In all cases, using  $\mathcal{P}_2 - \mathcal{P}_0$  elements gives much better results. As seen in Tables 3.1-3.2, mass conservation is obtained, and the FOSLS functional is still minimized, retaining its error approximation properties. Moreover, using unstable pairs of elements can even result in the divergence of the FOSLS functional. In the context of this problem, the solution is still obtained accurately, but the gradient of the solution is not captured well. The solution process is no longer minimizing the residual in the  $H^1$  norm.

In addition, the results show that the use of a coarse-grid improves the performance of the method. The second block in Table 4.1 shows results for performing one smoothing step of the subdomain solver with no coarse-grid correction. This does improve the conservation results, but not significantly. Performing 100 smoothing steps of the subdomain solver with no coarse-grid correction improves the mass conservation, but of course these iterations are expensive. Finally, the fourth set shows results for using one step of the subdomain solver with one solve on a coarse-grid. The mass conservation is retained and not much work is needed. Combining with the results from Table 3.1, this process requires around 4 iterations of MINRES for each local subdomain and for the coarse-grid. Each of these subdomains has less DOFs and, therefore, the work required to solve the constrained system there is a fraction of the cost of solving the original FOSLS system.

**5. Conclusions.** In summary, the results of this paper have shown that properties such as mass conservation can be obtained using the least-squares finite element method and a post-process subdomain correction method. There are many other

	$\mathcal{P}_1 - \mathcal{P}_0$				$\mathcal{P}_2 - \mathcal{P}_0$			
$1/h$	FOSLS							
	$m_L$	$\mathcal{F}$	$u_{err}$	$p_{err}$	$m_L$	$\mathcal{F}$	$u_{err}$	$p_{err}$
8	5.3e-2	1.65	0.223	0.070	5.5e-4	0.14	8.3e-3	1.2e-3
16	2.8e-2	0.90	0.134	0.019	3.7e-5	0.04	1.1e-3	1.4e-4
32	1.0e-2	0.48	0.053	0.005	2.4e-6	0.01	1.8e-4	1.7e-5
$1/h$	$N_{sd} = 9, M_s = 1, \text{ No Coarse Grid correction}$							
	$m_L$	$\mathcal{F}$	$u_{err}$	$p_{err}$	$m_L$	$\mathcal{F}$	$u_{err}$	$p_{err}$
8	2.6e-3	1.77	0.180	0.061	2.3e-6	0.14	8.4e-3	1.2e-3
16	1.3e-3	1.09	0.185	0.016	1.0e-7	0.04	1.2e-3	1.4e-4
32	2.9e-5	1.47	0.201	0.004	9.2e-9	0.01	1.8e-4	1.7e-5
$1/h$	$N_{sd} = 9, M_s = 100, \text{ No Coarse Grid correction}$							
	$m_L$	$\mathcal{F}$	$u_{err}$	$p_{err}$	$m_L$	$\mathcal{F}$	$u_{err}$	$p_{err}$
8	4.2e-12	1.83	0.184	0.059	1.0e-11	0.14	8.4e-3	1.2e-3
16	4.7e-8	1.12	0.181	0.015	9.1e-11	0.04	1.2e-3	1.4e-4
32	2.2e-1	7.81	0.413	0.005	4.5e-11	0.01	1.8e-4	1.7e-5
$1/h$	$N_{sd} = 9, M_s = 1, H = 2h$							
	$m_L$	$\mathcal{F}$	$u_{err}$	$p_{err}$	$m_L$	$\mathcal{F}$	$u_{err}$	$p_{err}$
8	8.8e-11	1.83	0.181	0.060	2.3e-13	0.14	8.3e-3	1.2e-3
16	1.3e-12	1.92	0.188	0.015	1.2e-13	0.04	1.2e-3	1.4e-4
32	1.1e-10	10.09	0.209	0.004	2.5e-14	0.01	1.8e-4	1.7e-5
$1/h$	$N_{sd} = 9, M_s = 1, H = 4h$							
	$m_L$	$\mathcal{F}$	$u_{err}$	$p_{err}$	$m_L$	$\mathcal{F}$	$u_{err}$	$p_{err}$
8	5.3e-3	1.84	0.191	0.060	1.2e-6	0.14	8.3e-3	1.2e-3
16	1.2e-3	1.20	0.186	0.016	1.8e-8	0.04	1.2e-3	1.4e-4
32	2.8e-4	2.63	0.200	0.004	3.0e-9	0.01	1.8e-4	1.7e-5
$1/h$	$N_{sd} = 9, M_s = 10, H = 4h$							
	$m_L$	$\mathcal{F}$	$u_{err}$	$p_{err}$	$m_L$	$\mathcal{F}$	$u_{err}$	$p_{err}$
8	3.9e-4	1.83	0.195	0.060	1.1e-7	0.14	8.3e-3	1.2e-3
16	4.0e-3	1.29	0.192	0.015	7.8e-9	0.04	1.2e-3	1.4e-4
32	4.3e-2	9.93	0.363	0.007	6.7e-10	0.01	1.8e-4	1.7e-5

Table 4.1: Mass loss, least-squares functional, and relative errors of solutions for  $\mathcal{P}_1 - \mathcal{P}_0$  elements (Left) and  $\mathcal{P}_2 - \mathcal{P}_0$  elements (Right).

methods, as mentioned in the introduction, that also improve conservation properties for least-square problems. These may involve reformulating the system or choosing better finite-element spaces for the original FOSLS system. For instance, nonconforming elements can be used that satisfy the mass conservation across interfaces much better than the standard polynomial spaces used here [1, 15, 16, 23]. The goal of our approach in the present paper is to show that the system can be solved as is, with

no alterations to the original FOSLS method. Thus, it should be considered a robust finite-element method for such systems which obtains physically accurate solutions efficiently. Care needs to be given in choosing the right spaces for the constraint system, so that a stable method is obtained and the FOSLS functional retains its important a posteriori error estimator properties. However, since this post-processing is done on local subdomains and/or on coarse-grids, only a fractional amount of computational cost is added to the solution process. Future work involves implementing the above algorithms in a multilevel way and including the coarse-space constraints in the local subdomain process. Also, other applications such as Stokes flow and MHD is worth considering.

## REFERENCES

- [1] G. A. Baker, W. N. Jureidini, and O. A. Karakashian. Piecewise Solenoidal Vector-Fields and the Stokes Problem. *SIAM Journal on Numerical Analysis*, 27(6):1466–1485, 1990.
- [2] T. Barth. On the role of involutions in the discontinuous Galerkin discretization of Maxwell and magnetohydrodynamic systems. In *Compatible spatial discretizations*, pages 69–88. Springer, New York, 2006.
- [3] M. Berndt, T. A. Manteuffel, and S. F. McCormick. Local Error Estimates and Adaptive Refinement for First-Order system Least Squares (FOSLS). *E.T.N.A.*, 6:35–43, 1997.
- [4] P. Bochev, Z. Cai, T. A. Manteuffel, and S. F. McCormick. Analysis of velocity-flux first-order system least-squares principles for the Navier-Stokes equations: Part I. *SIAM Journal on Numerical Analysis*, 35(3):990–1009, 1998.
- [5] P. Bochev and M. D. Gunzburger. Analysis of Least-Squares Finite-Element Methods for the Stokes Equations. *Mathematics of Computation*, 63(208):479–506, 1994.
- [6] P. Bochev, T. A. Manteuffel, and S. F. McCormick. Analysis of velocity-flux least-squares principles for the Navier-Stokes equations: Part II. *SIAM Journal on Numerical Analysis*, 36(4):1125–1144, 1999.
- [7] J.U. Brackbill and D C Barnes. The effect of nonzero  $\nabla \cdot \mathbf{B}$  on the numerical solution of the magnetohydrodynamic equations. *Journal of Computational Physics*, 35(3):426–430, 1980.
- [8] J.H. Bramble, T.V. Kolev, and J.E. Pasciak. A least-squares approximation method for the time-harmonic Maxwell equations. *Journal of numerical mathematics*, 13(4):237, 2005.
- [9] S. C. Brenner and L. R. Scott. *Mathematical Theory of Finite Element Methods*. Springer, 2nd edition edition, 2002.
- [10] M. Brezina, J. Garcia, T. Manteuffel, S McCormick, J. Ruge, and L. Tang. Parallel adaptive mesh refinement for first-order system least squares. *Numerical Linear Algebra with Applications*, 19:343–366, 2012.
- [11] Z. Cai, R. Lazarov, T. A. Manteuffel, and S. F. McCormick. First-order system least squares for second-order partial differential equations: Part I. *SIAM Journal on Numerical Analysis*, pages 1785–1799, 1994.
- [12] Z. Cai, T. A. Manteuffel, and S. F. McCormick. First-order system least squares for second-order partial differential equations .2. *SIAM Journal on Numerical Analysis*, 34(2):425–454, 1997.
- [13] Z. Cai and G. Starke. Least-squares methods for linear elasticity. *SIAM J. Numer. Anal.*, 42(2):826–842 (electronic), 2004.
- [14] G. F. Carey, A. I. Pehlivanov, and P. S. Vassilevski. Least-squares mixed finite element methods for non-selfadjoint elliptic problems II: Performance of block-ILU factorization methods. *SIAM Journal on Scientific Computing*, 16:1126–1136, 1995.
- [15] B. Cockburn, G. Kanschat, and D. Schötzau. A locally conservative LDG method for the incompressible Navier-Stokes equations. *Mathematics of Computation*, 74(251):1067–1095, 2005.
- [16] B. Cockburn, G. Kanschat, and D. Schötzau. A note on discontinuous galerkin divergence-free solutions of the navier-stokes equations. *Journal of Scientific Computing*, 31:61–73, 2007.
- [17] H. De Sterck, T. Manteuffel, S McCormick, J Nolting, J. Ruge, and L. Tang. Efficiency-based h- and hp-refinement strategies for finite element methods. In *Numerical Linear Algebra with Applications*, pages 89–114. Univ Waterloo, Dept Appl Math, Waterloo, ON N2L 3G1, Canada, 2008.
- [18] V. Girault and P. A. Raviart. *Finite Element Approximation of the Navier-Stokes Equations*. Springer-Verlag, revised edition, November 1979.



- [19] V. Girault and P. A. Raviart. *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms (Springer Series in Computational Mathematics)*. Springer-Verlag, August 1986.
- [20] J. J. Heys, E. Lee, T. A. Manteuffel, and S. F. McCormick. On mass-conserving least-squares methods. *SIAM Journal on Scientific Computing*, 28(5):1675–1693, 2006.
- [21] J. J. Heys, E. Lee, T. A. Manteuffel, and S. F. McCormick. An alternative least-squares formulation of the Navier-Stokes equations with improved mass conservation. *Journal of Computational Physics*, 226(1):994–1006, 2007.
- [22] J. J. Heys, E. Lee, T. A. Manteuffel, S. F. McCormick, and J. W. Ruge. Enhanced mass conservation in least-squares methods for Navier-Stokes equations. *SIAM Journal on Scientific Computing*, 31(3):2303–2321, 2009.
- [23] OA Karakashian and WN Jureidini. A nonconforming finite element method for the stationary Navier-Stokes equations. *SIAM Journal on Numerical Analysis*, 35(1):93–120, 1998.
- [24] P. D. Lax. *Functional analysis*. Wiley-Interscience, March 2002.
- [25] E. Lee and T. A. Manteuffel. FOSLL\* method for the eddy current problem with three-dimensional edge singularities. *SIAM Journal on Numerical Analysis*, 45(2):787–809, 2007.
- [26] T. A. Manteuffel, S. F. McCormick, J. Ruge, and J G Schmidt. First-Order System LL\* (FOSLL\*) for General Scalar Elliptic Problems in the Plane. *SIAM Journal on Numerical Analysis*, 43(5):2098–2120, January 2006.
- [27] S. MüNZenmaier and G. Starke. First-order system least squares for coupled Stokes-Darcy flow. *SIAM J. Numer. Anal.*, 49(1):387–404, 2011.
- [28] A. I. Pehlivanov and G. F. Carey. Error-Estimates for Least-Squares Mixed Finite-Elements. *Rairo-Mathematical Modelling and Numerical Analysis-Modelisation Mathematique Et Analyse Numerique*, 28(5):499–516, 1994.
- [29] A. I. Pehlivanov, G. F. Carey, and P. S. Vassilevski. Least-squares mixed finite element methods for non-selfadjoint elliptic problems I: Error analysis. *Numerische Mathematik*, 72:501–522, 1996.
- [30] J. P. Pontaza and J. N. Reddy. Space-time coupled spectral/least-squares finite element formulation for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 197(2):418–459, 2004.
- [31] Y Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial & Applied, April 2003.
- [32] G. Starke. A first-order system least squares finite element method for the shallow water equations. *SIAM J. Numer. Anal.*, 42(6):2387–2407 (electronic), 2005.
- [33] A. Toselli and O. B. Widlund. *Domain Decomposition Methods-Algorithms and Theory*. Springer Verlag, 2005.
- [34] S. P. Vanka. Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *Journal of Computational Physics*, 65(1):138–158, 1986.
- [35] P. S. Vassilevski. *Multilevel Block Factorization Preconditioners*. Matrix-based Analysis and Algorithms for Solving Finite Element Equations. Springer Verlag, October 2008.